

1           1. An application software program comprising an  
2 object-oriented, verifiable, type-safe and pointer-safe  
3 sequence of instructions residing on a computer-readable  
4 medium, wherein the program can be loaded to and executed by  
5 a resource-constrained device that is based on a processor  
6 architecture of fewer than 32 bits.

1           2. The software program of claim 1 wherein the  
2 program can be executed by a resource-constrained device  
3 based on a 16-bit processor architecture.

1           3. The software program of claim 1 wherein the  
2 program can be executed by a resource-constrained device  
3 based on an 8-bit processor architecture.

1           4. The software program of claim 1 wherein each  
2 instruction includes an 8-bit operation code.

1           5. The software program of claim 1 wherein the  
2 sequence of instructions is hardware platform-independent.

1           6. The software program of claim 1 wherein the  
2 instructions were converted from at least one Java class  
3 file and wherein at least some references to a constant pool  
4 were transformed to inline data.

1           7. The software program of claim 6 wherein the  
2 instructions comprise operation codes and operands and  
3 wherein at least some references to the constant pool are  
4 inlined into operands in at least some of the instructions.

1           8. The software program of claim 6 wherein the  
2 instructions comprise operation codes and operands and  
3 wherein at least some references to the constant pool are  
4 inlined into operation codes in at least some of the  
5 instructions.

1           9. The software program of claim 1 wherein the  
2 instructions can be executed by a virtual machine running on  
3 a microprocessor residing on the resource-constrained  
4 device.

1           10. The software program of claim 1 wherein the  
2 instructions can be executed on a portable smart card.

1           11. The software program of claim 1 wherein the  
2 instructions can be executed by a device that supports  
3 multiple data types, wherein the sequence of instructions  
4 includes data manipulation instructions, and wherein each  
5 data manipulation instruction is specific to a particular  
6 data type.

1           12. The software program of claim 11 wherein the  
2 data type associated with each data manipulation instruction  
3 is selected from among one of the following types: an 8-bit  
4 signed two's complement integer numeric type, a 16-bit  
5 signed two's complement integer numeric type and a 32-bit  
6 signed two's complement integer numeric type.

1           13. The software program of claim 11 wherein the  
2 instructions can be executed by a device that supports  
3 multiple reference types and wherein each reference type is  
4 selected from among one of the following types: a class  
5 type, an interface type and an array type.

1 14. The software program of claim 1 wherein the  
2 program includes at least one composite instruction for  
3 performing an operation on a current object.

1 15. An application software program comprising an  
2 object-oriented, verifiable, type-safe and pointer-safe  
3 sequence of instructions residing on a computer-readable  
4 medium, wherein the program can be loaded to and executed by  
5 a resource-constrained device having random access memory  
6 with a capacity of no more than about 64 kilo-bytes.

1 16. The software program of claim 15 wherein the  
2 program can be executed by a resource-constrained device  
3 having random access memory with a capacity of no more than  
4 about 4 kilo-bytes.

1 17. The software program of claim 15 wherein each  
2 instruction includes an 8-bit operation code.

1 18. The software program of claim 15 wherein the  
2 sequence of instructions is hardware platform-independent.

1 19. The software program of claim 15 wherein the  
2 instructions were converted from at least one Java class  
3 file and wherein at least some references to a constant pool  
4 were transformed to inline data.

1 20. The software program of claim 19 wherein the  
2 instructions comprise operation codes and operands and  
3 wherein at least some references to the constant pool are  
4 inlined into operands in at least some of the instructions.

1           21. The software program of claim 19 wherein the  
2 instructions comprise operation codes and operands and  
3 wherein at least some references to the constant pool are  
4 inlined into operation codes in at least some of the  
5 instructions.

1           22. The software program of claim 15 wherein the  
2 instructions can be executed by a virtual machine running on  
3 a microprocessor residing on the resource-constrained  
4 device.

1           23. The software program of claim 15 wherein the  
2 instructions can be executed on a portable smart card.

1           24. The software program of claim 15 wherein the  
2 instructions can be executed by a device that supports  
3 multiple data types, wherein the sequence of instructions  
4 includes data manipulation instructions, and wherein each  
5 data manipulation instruction is specific to a particular  
6 data type.

1           25. The software program of claim 24 wherein the  
2 data type associated with each data manipulation instruction  
3 is selected from among one of the following types: an 8-bit  
4 signed two's complement integer numeric type, a 16-bit  
5 signed two's complement integer numeric type and a 32-bit  
6 signed two's complement integer numeric type.

1           26. The software program of claim 24 wherein the  
2 instructions can be executed by a device that supports  
3 multiple reference types and wherein each reference type is  
4 selected from among one of the following types: a class  
5 type, an interface type and an array type.

1           27. The software program of claim 15 wherein the  
2 program includes at least one composite instruction for  
3 performing an operation on a current object.

1           28. A resource-constrained device comprising:  
2           memory for storing an application software  
3 program comprising an object-oriented, verifiable, type-safe  
4 and pointer-safe sequence of instructions;  
5           random access memory having a capacity of no  
6 more than about 64 kilo-bytes; and  
7           a virtual machine implemented on a  
8 microprocessor wherein the virtual machine is capable of  
9 executing the sequence of instructions.

1           29. The device of claim 28 wherein the  
2 microprocessor is based on an 8-bit architecture.

1           30. The device of claim 28 wherein the  
2 microprocessor is based on a 16-bit architecture.

1           31. The device of claim 28 wherein each instruction  
2 includes an 8-bit operation code.

1           32. The device of claim 28 wherein the sequence of  
2 instructions is hardware platform-independent.

1           33. The device of claim 28 wherein the instructions  
2 were converted from at least one Java class file and wherein  
3 at least some references to a constant pool are transformed  
4 to inline data.

1           34. The device of claim 33 wherein the instructions  
2 comprise operation codes and operands and wherein at least  
3 some references to the constant pool are inlined into  
4 operands in at least some of the instructions.

1           35. The device of claim 33 wherein the instructions  
2 comprise operation codes and operands and wherein at least  
3 some references to the constant pool are inlined into  
4 operation codes in at least some of the instructions.

1           36. The device of claim 28 wherein the virtual  
2 machine supports multiple data types, wherein the sequence  
3 of instructions includes data manipulation instructions, and  
4 wherein each data manipulation instruction is specific to a  
5 particular data type.

1           37. The device of claim 28 wherein the program  
2 includes at least one composite instruction for performing  
3 an operation on a current object.

1           38. A resource-constrained device comprising:  
2           memory for storing an application software  
3 program comprising an object-oriented, verifiable, type-safe  
4 and pointer-safe sequence of instructions; and  
5           a virtual machine implemented on a  
6 microprocessor that is based on an architecture of less than  
7 32 bits, wherein the virtual machine is capable of executing  
8 the sequence of instructions.

1           39. A resource-constrained device comprising:  
2           memory for storing an application software  
3 program comprising an object-oriented, verifiable, type-safe  
4 and pointer-safe sequence of instructions;

5 random access memory having a capacity of no  
6 more than about 64 kilo-bytes; and  
7 a processor capable of executing the sequence  
8 of instructions.

1 40. The device of claim 39 wherein the processor is  
2 based on an 8-bit architecture.

1 41. The device of claim 39 wherein the processor is  
2 based on a 16-bit architecture.

1 42. A resource-constrained device comprising:  
2 memory for storing an application software  
3 program comprising an object-oriented, verifiable, type-safe  
4 and pointer-safe sequence of instructions;  
5 random access memory having a capacity of less  
6 than about 64 kilo-bytes; and  
7 an application-specific integrated circuit  
8 (ASIC) capable of executing the sequence of instructions.

1 43. The device of claim 42 wherein the ASIC is  
2 based on an 8-bit architecture.

1 44. The device of claim 42 wherein the ASIC is  
2 based on a 16-bit architecture.

1 45. A smart card comprising:  
2 memory for storing an application software  
3 program comprising an object-oriented, verifiable, type-safe  
4 and pointer-safe sequence of instructions; and  
5 a virtual machine implemented on a  
6 microprocessor, wherein the virtual machine is capable of  
7 executing the sequence of instructions.

8           46. The smart card of claim 45 wherein the virtual  
9 machine is substantially a Java Card virtual machine.

1           47. The smart card of claim 45 wherein each  
2 instruction includes an 8-bit operation code.

1           48. The smart card of claim 45 wherein the sequence  
2 of instructions is hardware platform-independent.

1           49. The smart card of claim 45 wherein the  
2 instructions were converted from at least one Java class  
3 file and wherein at least some references to a constant pool  
4 are transformed to inline data.

1           50. The smart card of claim 45 wherein the  
2 instructions comprise operation codes and operands and  
3 wherein at least some references to the constant pool are  
4 inlined into operands in at least some of the instructions.

1           51. The smart card of claim 45 wherein the  
2 instructions comprise operation codes and operands and  
3 wherein at least some references to the constant pool are  
4 inlined into operation codes in at least some of the  
5 instructions.

1           52. The smart card of claim 45 wherein the virtual  
2 machine supports multiple data types, wherein the sequence  
3 of instructions includes data manipulation instructions, and  
4 wherein each data manipulation instruction is specific to a  
5 particular data type.



1           53. The smart card of claim 45 wherein the program  
2 includes at least one composite instruction for performing  
3 an operation on a current object.

1           54. A method of using an application software  
2 program including an object-oriented, verifiable, type-safe  
3 and pointer-safe sequence of instructions, the method  
4 comprising:

5                 receiving the software program in a resource-  
6 constrained device having random access memory with a  
7 capacity of no more than about 64 kilo-bytes; and

8                 executing the sequence of instructions on the  
9 resource-constrained device.

1           55. The method of claim 54 further including:  
2                 storing the sequence of instructions on the  
3 resource-constrained device.

1           56. The method of claim 54 further including  
2 accessing the software program over a computer network prior  
3 to downloading the program onto the resource-constrained  
4 device.

1           57. The method of claim 54 further including  
2 accessing the software program over the Internet prior to  
3 downloading the program onto the resource-constrained  
4 device.

1           58. The method of claim 54 further including:  
2                 transforming constant pool indices that appear  
3 in the received set of instructions to corresponding data  
4 values.